PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium

International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

Impressum
Published by

Publisher: Rector of the Ilmenau University of Technology
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff

Editor: Marketing Department (Phone: +49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

Faculty of Computer Science and Automation
(Phone: +49 3677 69-2860)
Univ.-Prof. Dr.-Ing. habil. Jens Haueisen

Editorial Deadline: 20. August 2010

Implementation: Ilmenau University of Technology
Felix Böckelmann
Philipp Schmidt

USB-Flash-Version.

Publishing House: Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilmenau

Production: CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

Order trough: Marketing Department (+49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher: Universitätsbibliothek Ilmenau
ilmedia
Postfach 10 05 65
98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

SIMULATING COMMUNICATION WITHIN A SATELLITE-BASED AUTOMATED TOLL COLLECTION SYSTEM

Karin Lunde, Lothar Kieble

University of Applied Sciences Ulm
Prittwitzstr. 10
D-89075 Ulm, Germany

ABSTRACT

In this paper, we describe a simulation framework to simulate a vehicle telematics system, using the example of a satellite-based toll collection system. The system consists of a large number of independent mobile agents which exchange messages with a single computing center, via a mobile communications network. The aim of the simulation study is the analysis of the frequency and the amount of communication, depending on system parameters and update strategies. The paper focuses on simulating communication caused by toll-collecting processes. As well as presenting the simulation framework and essential model components, it discusses several simulation results.

***Index Terms*—** Discrete-event simulation, mobility, OMNeT++, distributed agents, communication, telematics

1. INTRODUCTION

In several European countries, toll collection systems are in use which employ a great variety of technologies. The least invasive systems use satellite-based positioning systems like the Global Positioning System (GPS) to detect the position and the traveled distance of a vehicle on the highway network in order to calculate the corresponding toll fee. While vehicles are moving, communication may be caused by a variety of reasons, including toll-charging processes, software updates, or updates of geographic map data. As a rule, messages are exchanged via a mobile communications network. We have developed a simulation framework to explore the dependency of the frequency and the volume of the communication within such a satellite-based toll collection system on parameter values (such as debt limits and timeouts) and the choice of update strategies.

A similar system structure can be observed in a variety of distributed systems with a large number of mobile agents and a single computing center where messages or requests are processed, as is often the case in vehicle telematics applications. As an example application, we consider tolling commercial vehicles on the German highway network. Here, the implementation of the tolling scheme is of secondary importance seeing that the deployed detection technology (e.g. local map matching, local POI detection, or dumb clients transmitting pure GPS-traces which is questionable for data privacy reasons) dominates the kind and amount of transferred data. In Germany, heavy-duty commercial vehicles are, as yet, the only vehicles subject to road charges [1]. We assume that each vehicle carries an on-board unit (OBU), which is an on-board computer responsible for detecting toll events and for communicating with the central computing center (CC). The operating company states that in 2008 more than 650.000 OBUs were registered (see [2]). One of the challenges we are facing is this large number of distributed agents, which can not be reduced materially without distorting essential system properties due to the single computing center.

On-board units can be considered as mobile and distributed agents (see [3]). However, from our perspective, the simulation of communication in such a toll-collection system is not predominantly a multi-agent simulation, because essential features are missing: direct interactions between the vehicles, demanding intelligent decisions from the OBUs, are not the object of current analysis, and the internal logic of the on-board units can be described by a finite state machine (see Section 3). For this reason, we follow a discrete-event based modeling approach (see [4]).

In this paper, we focus on communication solely due to toll-collecting processes. To this end, we describe the OBU model and the communication protocol. Furthermore, we present some simulation results which were attained using OMNeT++ [5], and discuss a few tool-related issues.

2. THE SIMULATION FRAMEWORK

Since the toll-related state of an OBU changes at discrete moments in time and as the result of certain events, e.g. the completion of a toll road segment, discrete-event system simulation is used to analyze system behavior. Toll-related events depend heavily on the movement of the vehicles across the road network. To decou-

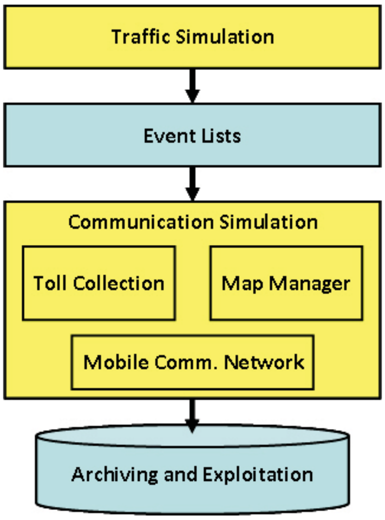

Fig. 1. The simulation framework.

ple the communication simulation from the traffic simulation, we defined an abstract interface between those two layers of our simulation framework.

In the first layer, traffic simulation generates trajectories of vehicles on the road network. The currently used algorithm is based on a random walk on the road network and is described in [6]. From the trajectories, a list is computed which contains events (each with a time stamp and a geographic position) in chronological order, like

- OBU `<ID>` activated or deactivated (ignition key);
- OBU `<ID>` entered or left toll road at segment `<ID>`;
- OBU `<ID>` completed toll road segment `<ID>`.

Owing to the generality of the interface, any algorithm can be used which produces time-stamped trajectories assignable to a vehicle ID and taking into account geographic constraints in consecutive tours. Mesoscopic or microscopic traffic simulations (for an overview, see e.g. [7]) allow the production of such trajectories as well, but they are still comparatively costly.

These event lists form the interface between the traffic simulation layer and the communication simulation layer which is implemented using OMNeT++.

3. COMMUNICATION DUE TO TOLL COLLECTING PROCESSES

Our model of an automatic road charging system consists of three entities which will be characterized in the following section (see Figure 2).

- The model of an **on-board unit (OBU)** integrates both the driver's behavior and the functionality of the on-board device. In Section 3.1, this central model component is explained in detail.

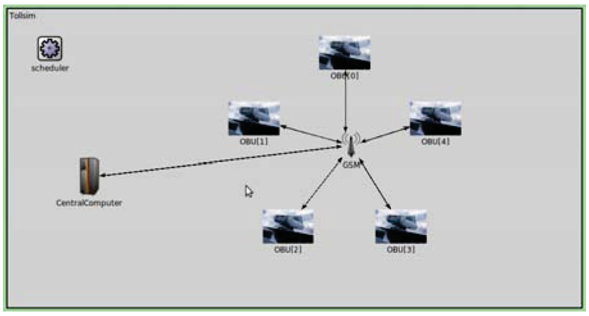

Fig. 2. A snapshot of OMNeT++, with a small number of OBUs.

- The **central computing center (CC)** receives messages sent by the OBUs via a mobile communications network, and returns a quittance message after receiving a valid collection message. The availability of the CC is modeled by means of a constant probability. A low probability indicates that the CC is frequently down for maintenance or has other problems.

- The **communication network** is used for data transmission between the mobile agents and the CC. For the purposes of the present analysis, it is modeled as a single global entity. We model the network availability with a constant probability. Each time an OBU has to send a message it checks whether the network is available. If not, the retry mechanism of the OBU is activated. In future, we intend to model the geographic distribution of the network cells, to be able to simulate regional blackouts of network parts.

3.1. On-board unit (OBU)

The most important entities in the modeled automatic road charging system are the on-board units, whose behavior is modeled by means of a finite state machine. The corresponding UML state diagram (Figure 3) shows the most important states as well as the events and conditions effecting state transitions.

At the start of the simulation, each OBU is in the state *Not Active*. The state of the OBU turns to *Active* when the OBU is activated through the ignition key. At the same time, the next route of the vehicle is planned. The state of the OBU turns back into *Not Active* as soon as the driver deactivates the ignition of the vehicle.

Within the state *Active*, there are three parallel sub state machines. Thus, the model is well-structured and open for future extensions.

3.1.1. Tour state machine

All events concerning (de-)activation and route planning have been generated by the traffic simulation in

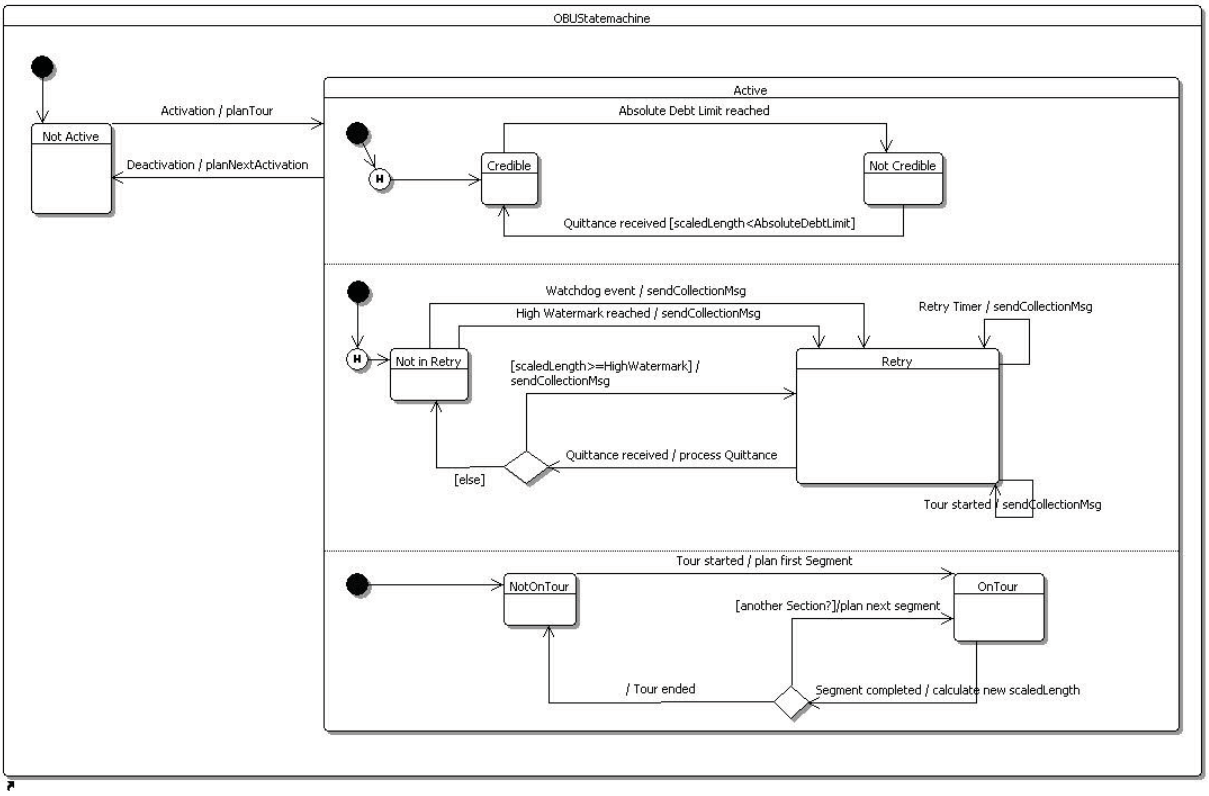

Fig. 3. Finite state machine modeling the behavior of an on-board unit.

a preprocessing step. The tour state machine handles the route planning events and calculates the credit sum, which is an important quantity for the other two sub state machines.

3.1.2. Retry state machine

The retry state machine encapsulates the sending of collection messages for billing purposes. In our model, the sending mechanism can either be triggered if the credit sum reaches the high watermark or if a watchdog timeout arises which means that the OBU has not communicated with the CC for a specific time interval. After a collection message has been sent, the OBU switches to state *Retry* and waits for incoming quittance messages from the CC. On receiving a quittance message, the collected credit sum is reduced by the acknowledged toll amount. If the resulting credit sum is still above the high watermark, another collection message will be sent.

In the case that the modeled OBU receives no quittance message within a given time interval, it attempts to send the last collection message again. This retry mechanism repeats infinitely unless the communication process succeeds and a quittance is duly received. To avoid duplicates in collection or quittance messages, we have introduced a sequence number, which is incremented each time the OBU receives a quittance message.

3.1.3. Credible state machine

The credibility state machine tracks whether the OBU is credible or not. The initial state is *credible*, which means the collected credit sum is below the absolute debt limit, and the vehicle can continue driving. If the credit sum exceeds the absolute debt limit, the OBU's state turns into *Not Credible* and the OBU ceases to take part in the toll-collection process. In reality, there is the alternative to pay the toll amount manually at the next toll-terminal (e.g. in a service area), but manual payment is not part of our model.

3.1.4. Model parameters

A central goal of this simulation study is the analysis of the system reaction to parameter changes. The global parameters of our system are the number of OBUs, the duration of the simulation and the probabilities which model the availability of the network and the CC. Of even greater interest are the parameters of the OBU state machine, namely the watchdog timer interval, the high watermark, the absolute debt limit, and the retry timer interval.

4. SIMULATING USING OMNET++

We chose OMNeT++ to implement the simulation model described above mainly for two reasons: firstly, it pro-

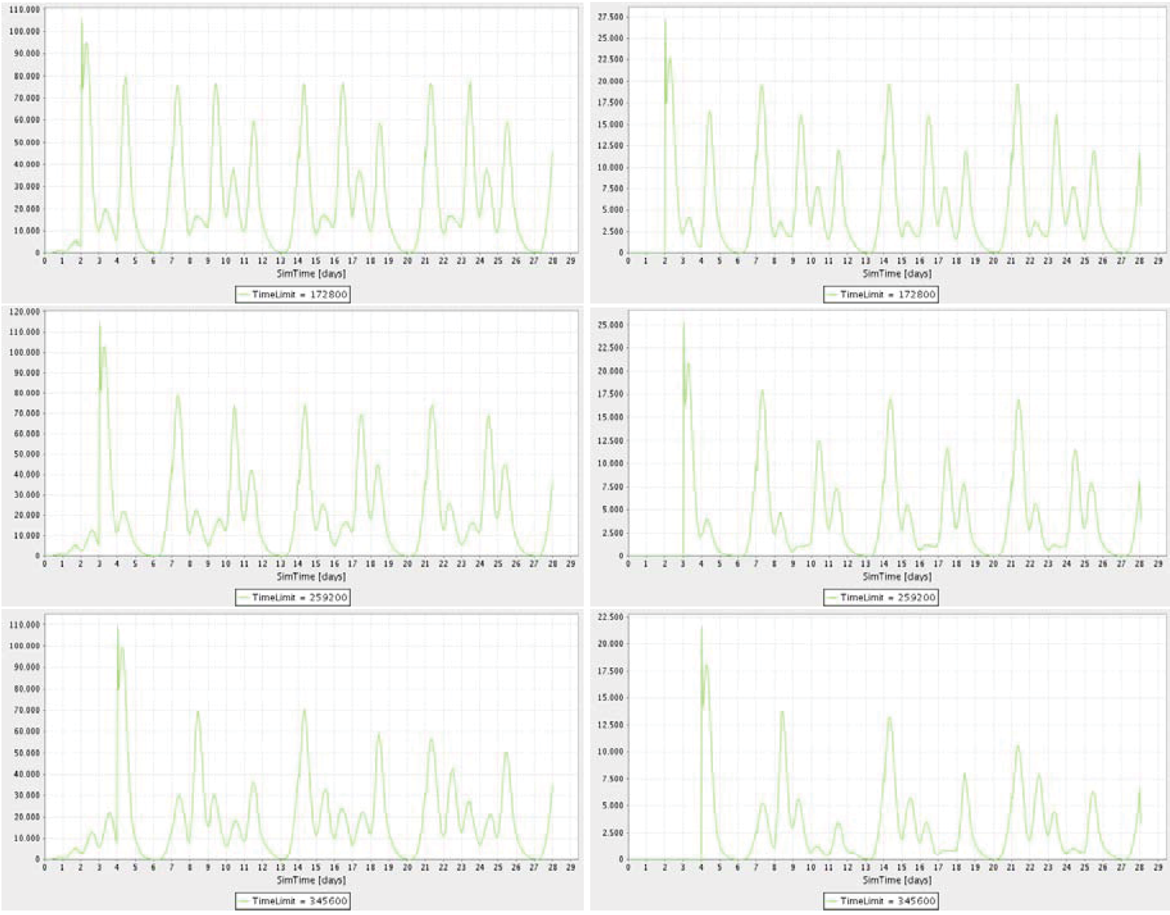

Fig. 4. Overall number of messages (left column) and number of watchdog events (right column) for different watchdog intervals (2 days, 3 days and 4 days).

vides a useful infrastructure for building models on a suitable level of abstraction, and secondly, it ensures a sufficiently high performance even if the simulation involves a large number of agents. Another advantage of OMNeT++ is that the entities are modeled in C++, which allows us to make use of object-oriented software design techniques, in particular design patterns.

Among the infrastructure provided by OMNeT++, there are two features we found especially useful. The finite state machines described in Section 3.1 are implemented by means of the FSM API. Furthermore, the configuration file mechanism of OMNeT++ allows the separation of the model from the parameters. This is essential for managing different parameter sets and the corresponding simulation experiments in a well-ordered way.

One challenge in implementing our simulation model was the large number of entities to be simulated. As a major issue, the interface to the traffic simulation was identified, especially with regard to memory consumption. Currently, the route generator is configured in such a way that it produces, on average, 15 segments per tour and 8 tours per vehicle and week. To simulate 500,000 OBU, the event count turns out to be about 60 million events per week. Instead of reading all events at the beginning of the simulation, we decided to just read windows of events which will be triggered in the near future, iteratively. To accomplish this, we have introduced another module called *Scheduler*. The *Scheduler* is triggered periodically by an event, reads the next group of events whose time stamps are below a given threshold, and distributes the read events to the OBUs (according to the vehicle ID). Because events are deleted immediately after they have been executed, this mechanism allows us to reduce memory consumption to a fraction of the memory which is needed if we read all events at the initialization phase.

5. RESULTS

All simulation experiments reported in this paper were executed on a PC with a dual core processor and 3 GB RAM. The operating system used is a 32 bit Ubuntu Linux.

One goal of the simulation study is to determine the influence of the model parameters on the communication behavior of the overall system. One of the most important performance measures is the overall number of messages sent by the OBUs, since each message implies expenses for the operating company. Figure 4 shows this overall message count, for an ensemble size of 500,000 OBUs and a simulation time of 4 weeks. The first thing which can be derived is that the simulation reaches a steady state after about two weeks. The number of sent messages is about zero at weekends in all cases, which is a consequence of the weekend truck ban which is part of the traffic simulation model. Another point which can be observed is that it depends on the chosen watchdog interval parameter on which days of the week the volume of communication reaches its peak level. Comparing the numbers of messages week by week shows that, in all kinds of settings, the maximum is observed on Mondays. The position of the next peak depends on the setting of the watchdog interval. When using an interval length of two days, a high amount of messages appear on Wednesdays and again on Fridays. Using a watchdog interval of three days causes a second peak on Thursdays, and a watchdog interval of four days causes a second peak on Fridays.

In this context, the behavior of the watchdog is also very interesting to investigate (Figure 4, right column). As mentioned before, the number of watchdog events on Mondays is very large in all cases. This is caused by the weekend truck ban whereby nearly all vehicles which start on Monday have not communicated with the CC since at least the previous Friday. Again the position of the next peak depends on the chosen watchdog interval. Correlating the number of watchdog events and the total number of messages sent it can be seen that the value of the watchdog interval timer has a direct influence on the amount of messages sent by the OBUs.

To sum up, it can be stated that if a very large watchdog interval is chosen, then communication is mainly caused by reaching the high watermark. On the other side, choosing a small watchdog interval leads to communication which is mainly caused by watchdog events. The value of the watchdog interval parameter also influences the total number of messages sent. In case of an interval length of two days the number of messages being sent after 4 weeks is 15.7 million, in case of an interval length of three days it is 13.7 million messages and in case of an interval length of four days it is 12 million messages. This indicates that the lower the chosen watchdog interval, the higher the number of messages which will be exchanged between the OBUs and the central computing center. Because the watchdog interval is meant to guarantee a deterministic charging process, a compromise has to be found which balances the amount of communication and the maximum time a client is allowed an open credit sum.

To investigate the resource consumption of our simulation model, we have run multiple simulation experiments with a varying number of OBUs and measured the duration of the experiments as well as the memory consumption (Figure 5). In these benchmarks, we have observed that both the duration of the simulation and the memory consumption grow linearly with the number of OBUs. Note that even with 500,000 OBUs, memory consumption remains below 2 GB, which enables us to execute our experiments on a standard PC.

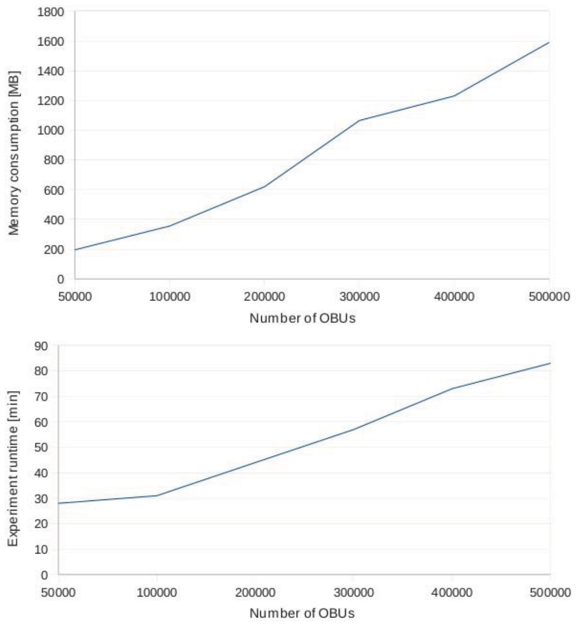

Fig. 5. Duration and memory consumption of simulation experiments

6. CONCLUSION

A simulation framework was designed which allows the simulation of frequency and the amount of communication in a vehicle telematics system. Simulation experiments with varying parameter values provide a deeper insight into the properties of the satellite-based toll collection system which was used as an example application.

In future, we intend to elaborate certain model aspects in more detail to cover simulation scenarios which have been so far disregarded, for instance, system recovery after a communications breakdown due to maintenance of the CC or failure of the communications network. In particular, we expect memory consumption will become an even greater issue. To overcome this difficulty, we are considering making more use of modern computer architectures.

7. ACKNOWLEDGMENTS

The work on this project was funded by Satellic Traffic Management GmbH. We thank our domain experts Dr. Marc-André Funk and Dr. Uwe Leinberger† for many stimulating discussions.

8. REFERENCES

[1] "Autobahnmautgesetz für schwere Nutzfahrzeuge in der Fassung der Bekanntmachung vom 2.Dezember 2004," BGBl. I, S. 3122, 2004.

[2] "Four years: Truck toll in Germany," Press release, see `http://www.toll-collect.de`, December 2008, Last visited 28.10.09.

[3] Wolf-Ulrich Raffel, *Agentenbasierte Simulation als Verfeinerung der Diskreten-Ereignis-Simulation unter besonderer Berücksichtigung des Beispiels fahrerloser Transportsysteme*, Ph.D. thesis, FU Berlin, 2005.

[4] Jerry Banks, *Discrete-Event System Simulation*, Prentice Hall, 2005.

[5] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *Proceedings of the European Simulation Multiconference (ESM'01), Prag*, June 2001.

[6] K. Lunde and F. Mast, "Ein Verkehrssimulator zur Simulation des Kommunikationsaufkommens in einem satellitengestützten automatisierten Mauterfassungssystem," in *Proceedings of ASIM'2009, Cottbus*, September 2009.

[7] G. Flötteröd, *Traffic-State Simulations with Multi-Agent Simulations*, Ph.D. thesis, TU Berlin, 2008.